

AN ABSTRACT OF THE THESIS OF

Albert L. Baily for the degree of Master of Science in

Electrical and Computer Engineering presented on August 19, 1991

Title: Multi-Media Presentation System Based On A Distributed
Control Network

Redacted for Privacy

Abstract approved: —

James H. Herzog

Control oriented local area networks (COLANs) are being installed in factories at an increasing rate. Traditionally, process control has operated with a master scheduler (computer) monitoring a number of points in a control grid. As the complexity of today's process control needs grow, the need to process information locally increases. Microcontrollers, networked with a master scheduler, can collect data from a locus of points and make decisions as to whether the master needs to be notified or not. By processing data locally, memory and execution time are freed up for the master scheduler. Task implementation becomes modular in nature, resulting in process control software that is easier to write, and

maintain.

This structure is the basis for COLAN V, a low cost, real-time, distributed control network developed at Oregon State University. COLAN V was used as the foundation for the creation of a multi-media presentation system. Six microcontrollers were networked together to remotely control the operation of projectors, projector screens, and lighting. Based on the application the master scheduler was replaced by a tape player. This allowed the storage of the audio part of the presentation on one track of the tape and the storage of the synchronized control signals on the other track. This distributed control network supplied a low cost solution to a need that was not addressed by the commercial market at any price.

Multi-Media Presentation System
Based On A Distributed Control Network
by

Albert L. Bailly

A THESIS

Submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of
Master of Science

Completed August 19, 1991

Commencement June, 1992

APPROVED:

Redacted for Privacy

Associate Professor of Electrical and Computer Engineering in charge of major

Redacted for Privacy

Head of Department of Electrical and Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented _____ August 19, 1991 _____

Formatted by Valerie Young

TABLE OF CONTENTS

1.	INTRODUCTION	1
	1.1. Background	1
	1.2. Thesis Objective	2
2.	NETWORK STRUCTURE	6
	2.1. COLAN V Network Structure	6
	2.2. Modifications to COLAN V	7
3.	TASK ORIENTED CONTROL STRUCTURE	8
	3.1. Operating System Structure	8
	3.2. Command Packet Format	8
4.	NETWORK HARDWARE	11
	4.1. Tape Deck	11
	4.2. Modem	11
	4.3. Personal Computer	12
	4.4. Network Interface Units(NIU)	12
	4.5. Transmission Medium	16
	4.6. Task Nodes	16
5.	APPLICATION SPECIFIC HARDWARE	21
	5.1. Projector Controller Hardware	21
	5.2. Light Control Hardware	27
6.	HIGH LEVEL SOFTWARE	29
7.	TASK NODE SOFTWARE	30
	7.1. Task Master Operating System	30
	7.2. Task Node Internal Memory (256 Byte)	31
8.	TASK SOFTWARE IMPLEMENTATION	35
	8.1. Global Task	35
	8.2. Application Specific Tasks	37
	8.3. Algorithm For Light Control	40
	8.4. Pseudocode For the Fade Algorithm	44
9.	PRESENTATION TAPES	47
	9.1. Presentation Tape Creation Equipment	47
	9.2. Creating Taped Presentations	48

10.	OPERATION OF PRESENTATION SYSTEM	54
	10.1. Operational Components	54
	10.2. Operation of the Light Controller	55
	10.3. Operation of the Projector Controller	56
11.	MAINTENANCE	57
	11.1. Tasks not Executed by any Nodes	57
	11.2. Communications OK but Lights/Lamps Fade Incorrectly	59
	11.3. Lights Stay On/Off all the Time	60
	11.4. Projector Lamps Stay On\Off all the Time	61
12.	CONCLUSION AND RECOMMENDATIONS	62
	12.1 Conclusions	62
	12.2 Recommendations for Future Research	63
13.	BIBLIOGRAPHY	66
14.	APPENDIX	68

LIST OF FIGURES

Figure 1.1 Museum Distributed Network Configuration	5
Figure 4.1 Network Interface Unit (NIU)	14
Figure 4.2 Minimum Component Task Node	18
Figure 5.1 Control Relays To Slide Projector Interface	23
Figure 5.2 Zero Crossing Detector Circuit	25
Figure 5.3 Projector Screen Relays	26
Figure 8.1 Museum Lighting Assignments	38
Figure 8.2 Voltage Time Area Power Analysis	42

Multi-Media Presentation System Based On A Distributed Control Network

CHAPTER 1

INTRODUCTION

1.1. Background

A local area network (LAN) is defined as any network of computers that are located within about 1 km of each other. The transmission medium, usually twisted pair wire or coaxial cable, is common to all machines. This allows the sharing of information and more efficient use of the resources available to all computers connected to the LAN. As the needs of industry become more complex LANs are being installed at an increasing rate, in many diverse applications [1].

A distributed control LAN is, by definition, a LAN used to control some process. The typical network has a distributed set of microcontrollers, each with a dedicated function, and a system scheduler, or host, that monitors and controls the activities of the network. The host acts as the user interface to the network, providing a 'window' to view, and control, the operations of the entire network from a single location. Typically the host is some sort of personal computer (PC) running a software program that performs this interface. A microcontroller usually refers to a microprocessor designed for control applications. A standard microcontroller

would be typified by an 8 bit architecture, direct input-output (I/O) processing ability, and a limited but control oriented instruction set.

The use of microcontrollers in a distributed system reduces the necessary bandwidth on the communication bus by processing the collected data locally. Rather than transmitting all data collected to the host for processing, as would a typical data acquisition node, the microcontroller can filter the data and make decisions locally, based on its embedded program. As control systems become more complex the need for increased communication bandwidth makes microcontrollers a viable alternative [6]. In 1990 the COLAN V (Control Oriented Local Area Network) was defined. This network evolved from the original 'Taskmaster' distributed control system created in 1987. COLAN V was based on a host computer (IBM PC) monitoring one or more task (microcontroller) nodes. The nodes were based on the SIBEC II, a small board computer designed by Binary Technology. The microcontroller used by the SIBEC II is the Intel 8051.

1.2. Thesis Objective

The Paul Jensen Museum at Western Oregon State College was in need of a control system to operate a one room diorama. The system was to run in conjunction with a recorded cassette tape explaining the exhibits. The immediate requirements of the museum were to control lighting, projectors and projector screens. The system had to be robust, flexible, and easily operated by a typical non-technical volunteer worker.

A Composer system with the ability to synchronize a taped presentation with the control of lights, projectors, and screens (if it exists) would be extremely costly. Additionally, the composer system would have to possess the ability to grow, in ways as yet undetermined, with the needs of the museum (i.e. VCR control, motor control, temperature control, etc.).

A distributed control network provides the best/only solution to the problem. Adding additional functions to the system requires only adding another node to the network. Additional functions can be added, or deleted, from the control network as independent modules resulting in a highly flexible system. It was this fact that motivated the design and implementation of the distributed control network that is the basis of this thesis. The network design is based on COLAN V, the most recent of the series of control oriented networks, developed at Oregon State University.

The implementation of the network required:

- 1) Modification of the system operating system to reduce chip count, simplify register usage, and increase operating speed.
- 2) Design of minimum chip task node to reduce costs of implementation (17 chips reduced to 5).
- 3) Design and Implementation of projector screen controller and light controller.
- 4) Modifications to COLAN V task library and hardware necessary for the implementation of this application.

This system was designed and installed in the Summer of 1991.

Figure 1.1 depicts the distributed control network for the museum diorama. The following is written to give insight into system design and to aid those in the maintenance and expansion of the network. This network is expected to be in use for many years in a real world environment. This paper will attempt to address those areas that are critical to its operation and maintenance.

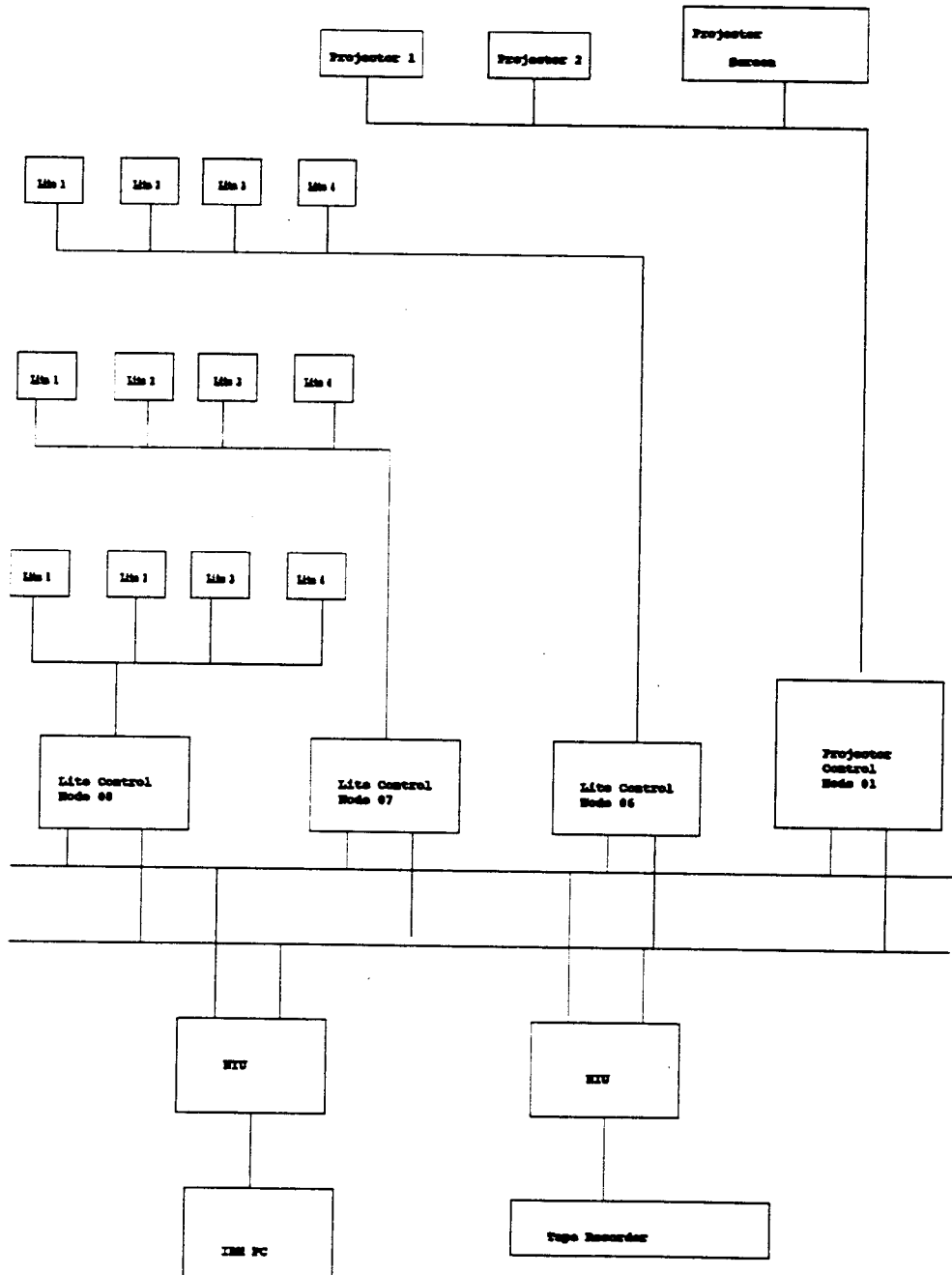


FIGURE 1.1: MUSEUM DISTRIBUTED NETWORK CONFIGURATION

CHAPTER 2

NETWORK STRUCTURE

2.1. COLAN V Network Structure

The COLAN V network consists of a host computer, RS-485 communication bus, and the task nodes.

The host computer could be any machine that supports RS-232 communication. Typically this has been an IBM compatible machine (PC). The host controls the operation of the task nodes by sending short command packets (2 to 18 bytes). Both individual and global addressing of the nodes is possible. The host is the master scheduler, but any node can gain access to the bus dependent on that particular node's task software. RS-485 was selected due to its being a truly multipoint bus that is both robust and economical. The task nodes, and host computer, gain access to the bus through a Network Interface Unit (NIU). The NIU translates between RS-232 and RS-485 for each node on the network. The transport medium is twisted-pair wire. Due to its simplicity, and the fact that the bus is only lightly used, CSMA (Carrier Sense Multiple Access) was selected as the communication protocol. The task nodes are Intel 8051 small board computers running under the Taskmaster operating system.

2.2. Modifications to COLAN V

Since the museum project required synchronization of a taped audio presentation and network control signals, utilization of the two tracks on the recorded tape seemed logical. If the audio portion of the presentation (music, narration, animal sounds, etc.) were placed on only one track of the tape then the other track, if the baud rate was restricted to the audio range, could be utilized for control signals. Under this restriction the system controller was replaced by a tape recorder and modem. The origin of replacing the system controller with a tape deck was pioneered by COLAN V, though the results were somewhat compromised. The PC was relegated to the construction of command packets and the downloading of them, through the modem, to a multi-track recorder. The PC can also be used to monitor the bus, and as an aid in system debugging by replacing the tape deck during testing. The task nodes were converted directly to RS-485 reducing the chip count by four for each node. The NIUs were still required for the PC and the modem, which are RS-232 based. The microcontroller was upgraded to the Intel 8052. The 8052 is a more powerful version of the 8051 with enough internal memory to allow the task queue to be moved to the internal RAM while maintaining compatibility with the COLAN V command packet format. This removed the need for the external RAM, allowed tighter code, and increased operating speed.

CHAPTER 3

TASK ORIENTED CONTROL STRUCTURE

3.1. Operating System Structure

The network is composed of a master scheduler, either a PC or a tape deck in this case, and associated task nodes. The nodes execute tasks based on the command packets sent by the scheduler. Each node has a unique address and, though each packet is received by all nodes, only if the node address matches that of the command packet will they execute the transmitted task.

The operating system software resides in each node and remains in the background during task execution. The operating system is responsible for receiving packets, sorting out those addressed to it, updating the task queue, initiating tasks, managing the tasks, and terminating tasks.

3.2. Command Packet Format

A command packet describing a task is composed of five data fields enclosed by a set of flags. The left and right braces are the flags used to denote the beginning and the end of a packet. The meaning of each field is inferred by its position in the packet. Blanks are used only to separate the fields and for clarity.

A typical packet has the following format [2]:

{ AA P NN S DD }

The individual fields in the packet shown above have the following definitions:

- { Packet start flag. This character denotes the start of the packet.
- AA Destination node address. This field consist of two ASCII characters indicating the destination of the packet. The special address "00" is reserved as the universal address that directs the packet to all nodes on the network.
- P Pre-Execution control character. This single character describes the method to process the task. There are three permissible characters:
 - : Queued Task. The task is to be placed at the end of the Task Queue and only to be executed at the completion of all the preceding tasks on the queue.
 - ? Synchronized Task. Same as Queue Task except the task will not start executing until the host issues the special synchronize task.
 - ! Immediate Task. The task is to be executed immediately.

If a queued task is currently running the queued task is temporarily suspended and will resume upon the completion of the immediate task. An immediate task can not interrupt a currently running immediate task and will be unceremoniously dumped.
- NN Task number. The two ASCII characters specifies the task to be performed.

- S Post-Execution control character. This single character describes the operation to be performed on the packet after it has been executed.
- . Execute Once Task. The task is to be executed only once and is discarded after execution is completed.
- + Re-Queued Task. Place the task at the end of the Task Queue after execution is completed.
- DD Data field. This field may have up to 5 pairs of hexadecimal characters. This field is used to pass arguments from the host.
- } Packet end flag.

Minimal error checking is done for each packet. The task writer is responsible for maintaining compatibility with the given format. If a received packet is not terminated with the 'packet end flag' ('}') the command will not be executed and will be dumped by the operating system when the next packet start flag ('{') is received.

CHAPTER 4

NETWORK HARDWARE

The network hardware for the museum project consists of a tape deck, modem, PC, NIU, transmission medium, and the task nodes.

4.1. Tape Deck

The tape deck is a standard product and its operation will not be covered here. The left channel contains the audio track and the right contains the control signals. The control tapes were made by using a "MR-200MK II" multi-track recorder. Standard tape decks have too much cross-talk between channels and should not be considered to create a control tape. The left channel output is connected to the audio system amplifier. The right channel output connects to the receiver (input) of the modem. The inputs are not used as the tape lacks the ability to respond to the task nodes.

4.2. Modem

Since the bandwidth necessary was minimal, and the tape player can not deal with phase modulation encoding, the baud rate of 300 bps was selected. Frequency shift keying (FSK) is used by standard 300bps modems and has a base frequency of 2000 Hz. For reproduction with a tape recorder this is ideal. 300 baud modems have been rendered obsolete by advancing

technology and can be purchased for next to nothing (If you can find them).

A 300 baud modem was obtained to replace that designed in COLAN V. To adapt the modem for use on the network it was necessary only to disconnect the wires to the speakers in the mouthpiece and ear-piece and connect these wires to RCA jacks mounted on the side of the modem case.

The modem is a DCE so the the cable connecting the modem to the NIU must be a null modem (flip pins 2 & 3). The settings to download to the network from the tape are F (full duplex) and O (originate).

4.3. Personal Computer (PC)

While the tape deck is acting as the system host a PC is necessary to create presentation tapes and to maintain the network. For either purpose the PC must have a serial port. For this application the PC chosen was an IBM compatible machine.

To facilitate the creation of presentation tapes the PC needs a software interface. This interface must be able to down load command packets at controlled intervals. The interface currently used is MUSE.BAS.

For system development, and debugging, it is necessary to add the PC to the network. To do this the PC must have the ability to emulate a terminal in order to monitor the RS-485 bus and to down load packets for testing.

4.4. Network Interface Units (NIU)

The NIUs are RS-232/RS-485 converters. As defined by the EIA RS-232

is a single ended data transmission system not practical for network applications (though some daisy chained systems do exist). To meet the needs for true multipoint communications, the EIA established RS-485 in 1983. This standard allows up to 32 driver/receiver pairs to be connected to the bus. RS-485 is a differential transmission system. Communication is based on the difference between two signal lines. This yields high common mode rejection of unwanted signals due to noise or ground shifts. Data rates range from 10Mb/s at 40 ft. to 100kb/s at 4000 ft. If these data rates are being approached care must be taken to use twisted pair cable as the transmission medium and terminate transmission lines with 120ohm resistors.

In COLAN V each node had its own NIU. For even a small network this becomes a significant expense in terms of cost and space. The solution was to design nodes based on RS-485 and only deal with RS-232 when necessary. Being RS-232 based, only the modem and the PC require NIUs. Figure 4.1 shows the schematic for the NIU. The components of the NIU are the MAX232 transceiver, the RS-485 75174 (driver)/75175 (receiver) pair, and a 74LS123 dual monostable multivibrator.

The MAX232 is a TTL/RS232 transceiver. It acts as the interface between RS232 devices and the RS485 devices which require TTL inputs. The MAX232 meets all EIA RS-232 specifications while utilizing a charge pump system allowing it to operate on a single +5 volt supply.

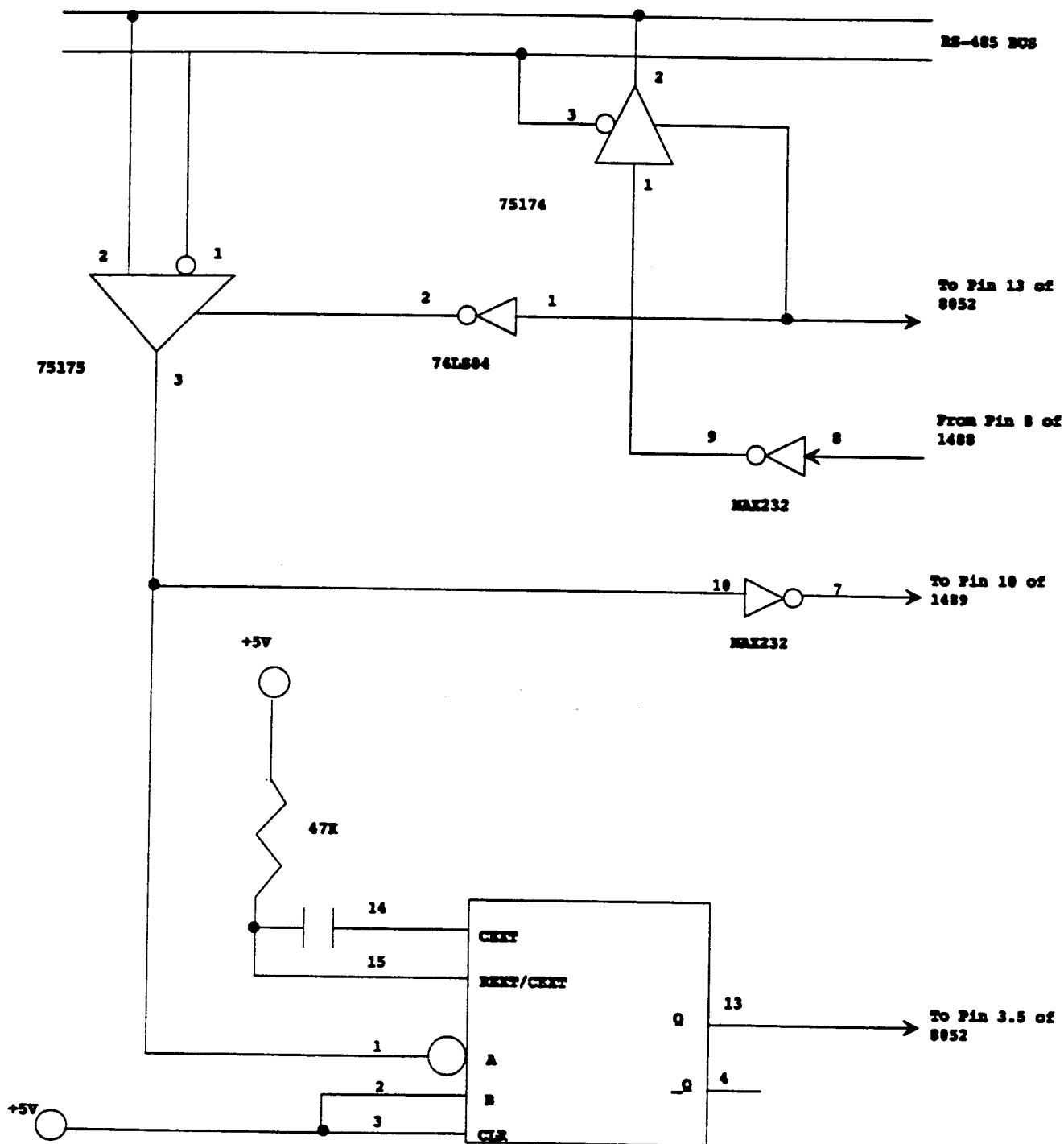


FIGURE 4.1: NETWORK INTERFACE UNIT (NIU)

The 75174/75175 RS-485 driver/receiver pair meet IEA standards. They features -7volt to +12volt common mode range yielding high noise immunity. The receiver has high input impedance (12kohm minimum) and 200mvolt hysteresis, again reducing the affects of noise. The drivers can withstand bus contention and bus faults, but must be in the tri-state mode when not transmitting or bus loading will garble communications. This is accomplished by driving the enable pins with complemented signals, thus simultaneous transmitting and receiving are not allowed. This pin is driven by the 8052 and is controlled by software. Typically the driver is disabled until transmission is necessary.

Note that if the modem and PC are connected to the bus at the same time, neither one has code or circuitry to disable its RS-485 driver. A manual switch would be a quick fix.

The 74LS123 is a typical monostable multivibrator. The length of the output pulse is controlled by the RC pair tied to pins 14 and 15. The function of this device is to provide the "sense" for the CSMA protocol used to monitor the bus. The one-shot is fired by the output of the 75175. As long as data bits continue to be received the one-shot will be retriggered. As long as the RC time constant is longer than the transmission time for a byte of data the output pulse will signal that the bus is in use. This pulse is fed to a pin on the 8052 and is checked by software prior to transmission. The RC time constant can be different for each board, giving a method to prioritize bus access. While the RS-485 interface has been moved to the task

nodes this protocol remains intact.

4.5. Transmission Medium

Twisted pair cable and phone cable provide the transmission medium for the network.

The phone cable is used for the short runs. It is color coded, has easily implemented connections, and has four wires allowing +5volts and ground to be based on the unused two lines. This eliminates the need for separate power supplies for the necessary NIUs. The color code is Black = Ground, Yellow = +5, Green = Y or (A), and Red = Z or (B).

Twisted pair should be used for long runs or for high baud rates. Twisted pair cable is inexpensive, readily available, and has good noise immunity. Noise in transmission lines is proportional to the impinging magnetic fields and the loop area of the wires. Twisted pair not only reduces the area between the wires but the twist reverses the alignment with the magnetic field. This causes the currents in each half loop to cancel those in the next half loop reducing noise on the line. Another desired effect is that twisted pair tends to balance the impedance of the line [NAT 89]. For many applications twisted pair is the best solution. For high baud rates use 120ohm terminations for impedance matching.

4.6. Task Nodes

The Task Nodes are the heart of the system. The processor used for this application is the Intel 8052, but it should be noted that any processor

with serial communication could be used. The only requirement is that it be compatible with the Taskmaster operating system. Here the 8052 will be used as a basis for the task node (figure 4.2).

The minimum number of chips for a complete task node is only five.

This results in very simple and economical designs that can accomplish a surprising number of fairly complex tasks. For projects that are more demanding resources can be added, such as additional ROM, RAM, and I/O devices. The 8052 features [4]:

1. 8-bit CPU optimized for control applications
2. Extensive Boolean processing capabilities
3. 64K Program Memory address space
4. 64K Data Memory address space
5. 8K bytes of on-chip ROM (8752 version)
6. 256 bytes of on-chip Data RAM, upper 128 bytes are addressed indirectly
7. 32 bidirectional and individually addressable I/O lines
8. Three 16-bit timer/counters
9. Full duplex UART
10. 7-source/6-vector interrupt structure with two priority levels (expandable to three levels with software)

This particular project has a baud rate of 300 bps, but the 8052 (if clocked by the standard 11.0952Mhz X-tal) supports all standard rates up to 19.2k baud.

Each node has port pin p1.0 dedicated to the "Friendly Blink". This is an LED that is blinked by the Taskmaster operating system whenever the 8052 is reset. It can be used as a quick test of the RS-485 network by

transmitting the reset command ('{%}').

COLON V included a dip switch input to set/change the node address and baud rate. For this particular application the dip switch was abandoned and the baud rate and node addresses were hard coded.

CHAPTER 5

APPLICATION SPECIFIC HARDWARE

The type and number of the "things" to be controlled define the number of task nodes necessary in a control network. For this project it was necessary to control two projectors, a projector screen, and up to 16 lights. It was determined that one board could control the projectors and screen, and four boards could control the lights. Six boards were built, based on the five chip design. These were to provide the four light control nodes plus two boards for maintenance. A seventh board was purchased from 'Binary Technology' for the projector controller. This board makes use of the popular 8255 I/O chip.

5.1. Projector Controller Hardware

The specific hardware for the projector controller included:

Potter & Brumfield SSR and SSRQ series solid state relays

A zero cross detector circuit

The SSR and SSRQ relays (figure 5.1) were used to provide an independent, optically isolated interface between the projector and the 8255. Two types of the relays are used in the implementation, random voltage turn-on (SSR) and zero voltage turn-on (SSRQ). The SSR relay provides full phase control. Phase control is a method of applying the ac supply to the

load for a controlled fraction of each cycle. Zero voltage turn-on only allows the relay to be on for an integral number of half cycles.

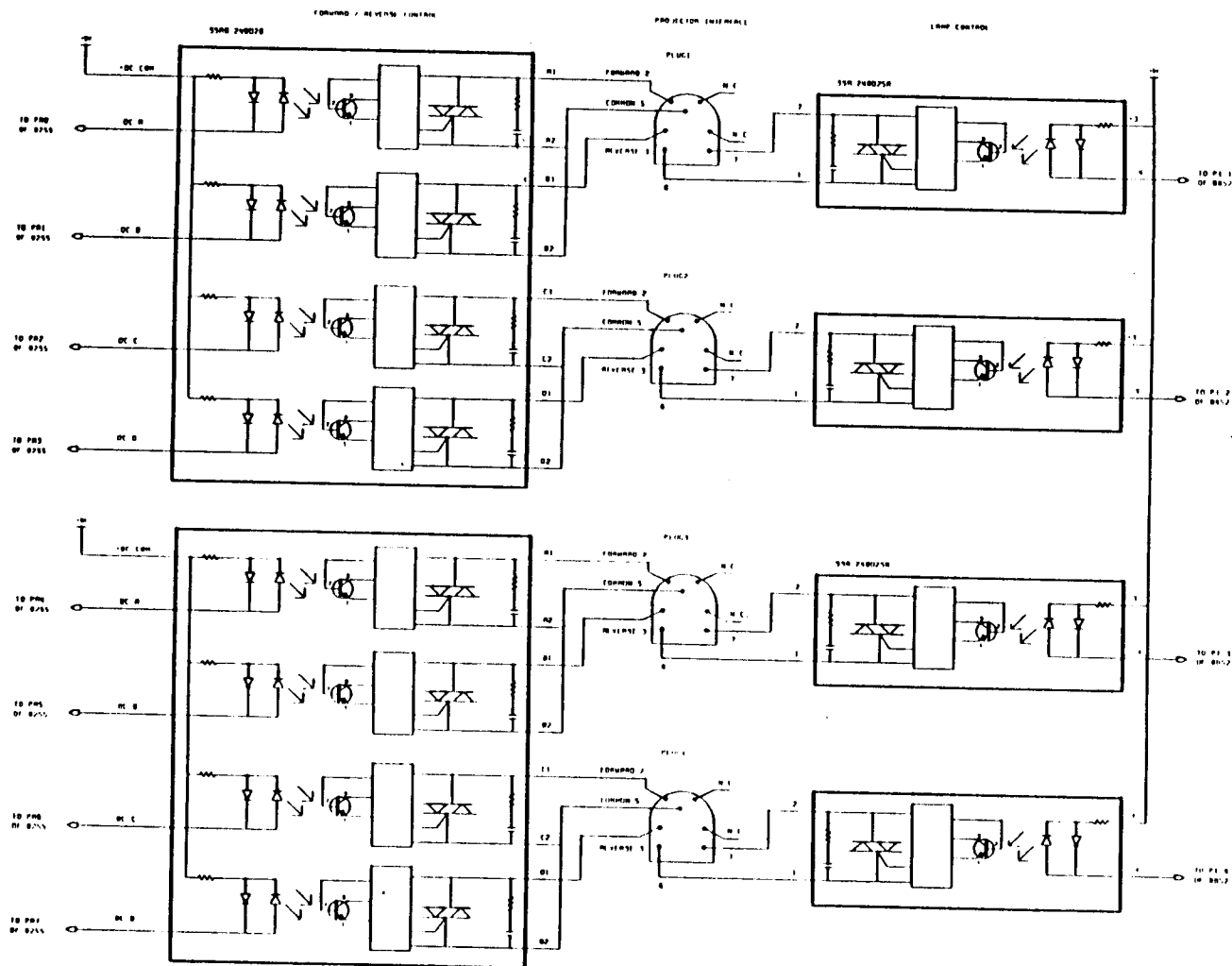


FIGURE 5.1: CONTROL RELAYS TO SLIDE PROJECTORS INTERFACE

The zero-crossing detector circuit (figure 5.2) was designed with a LM311 comparator and a opto-isolated npn transistor (NTE3041). The input to the comparator was current limited by the 100kohm resistor and voltage limited by the back to back diodes. Each time the 60Hz input crosses the 0 voltage axis the transition is sensed by the comparator and a synchronous square wave results at the open collector output, driving the NTE3041. A transition occurs 120 times a second. This transition is used as a reference for timed control of the SSR relays and as a time base for the rate that fading will occur. The output of the opto-isolator inputs to a port pin of the 8052.

The projector screen controller hardware (Figure 5.3) was designed with simple 12 volt SPDT relays. Based on the screen motor operating current of 3.5amps, 10amp relays were selected. The relays used were purchased at Radio Shack (Cat. No. 276-248). The need to provide user control when the network was not in use, and the nature and wiring of the installed wall switch (SW1) defined the function and wiring of the relays. SW1 is a DPDT switch with switch positions on-open-on . The relays work in parallel with the switch.

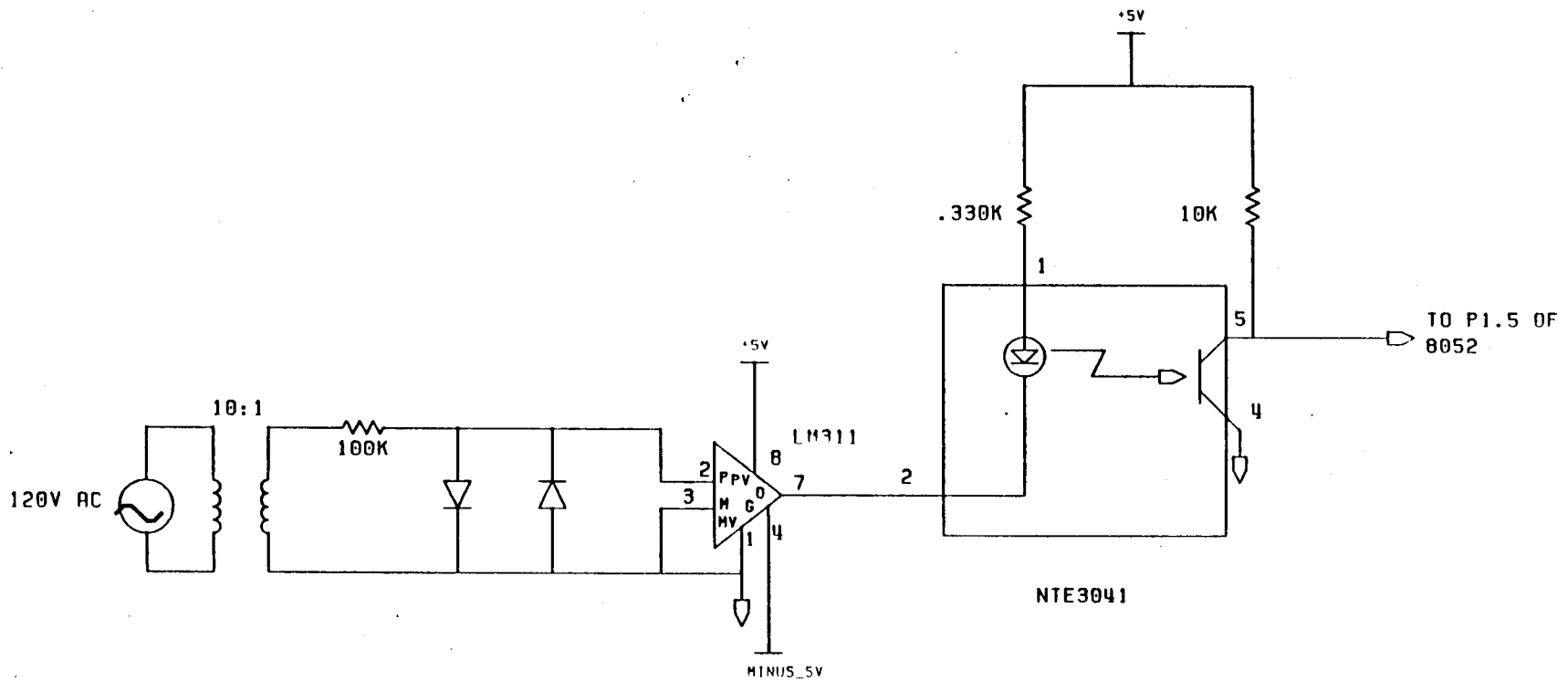


FIGURE 5.2: ZERO CROSSING DETECTOR CIRCUIT

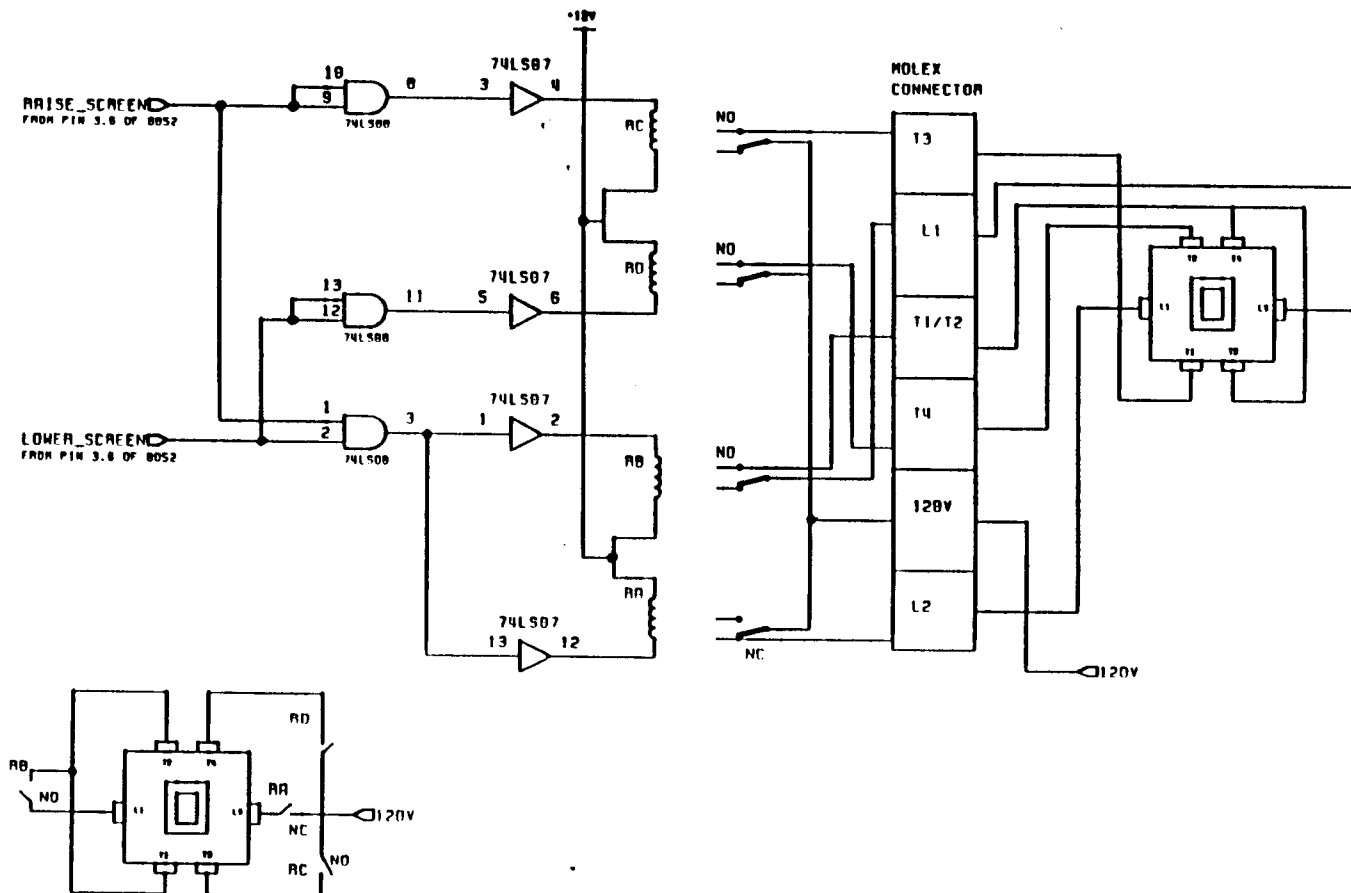


FIGURE 5.3: PROJECTOR SCREEN RELAYS

RA is normally closed when the screen controller is not driving the motor. This allows SW1 to operate normally when the network is not driving the screens motor. When the motor is being driven by the microcontroller this relay is opened as protection against shorting the motor windings by the inadvertent throwing of SW1. T2 and T1 are wired together so RB, which is normally open, is closed whenever the controller either raises or lowers the screen. RC is closed whenever the raise screen command is issued and RD when the lower screen command is received. The on current for the relays is 38ma maximum. The 74LS07 open collector drivers (sink 40ma maximum) were selected. The 74LS08 was selected to provide the ANDing of the active low signals from INT1 and T1 of the 8052. The free wheeling diode circuits are used to protect the open collector outputs of the 74LS07 from the energy released by the collapse of the relay's magnetic fields at turn off [H&H]. The logic equations for the four relays follow:

$$RD = T1, \overline{RD} = \overline{INT1}, \overline{RB} = \overline{RA} = T1 \text{ OR } INT1 \quad (5.1)$$

5.2. Light Control Hardware

The specific hardware for the light controller included:

400V 6 amp triacs

Opto-isolators with triac outputs (MOC3010)

Opto-isolator drivers (74LS07)

Zero cross detector

Isolation transformer

Triacs can be thought of as back to back Silicon Controlled Rectifiers (SCRs). When current is pulsed to the gate the device conducts, and will continue to conduct until the current through MT1 and MT2 is 0. This occurs at every zero-crossing. The triacs were selected to allow plenty of headroom for safety and future expansion. Note, the gate is connected through the opto-isolator to MT2.

The MOC3010s are diode input, Triac output opto-isolators. When the diode is turned on the triac output functions like the triac above. The MOC3010 offers 2500volts isolation between the high voltage and the digital logic.

The 74LS07 are high current open collector buffers. They can sink a maximum of 40ma.

The zero cross detector is exactly like that of the projector controller. It was apparent that if the light controllers were located physically close together one detector could service all the boards.

The isolation transformer was used to step down the 120volt input to 12 volts for the input to the zero cross detector. If a step down transformer is used care must be taken as the output of the transformer may very well lag the actual 120 ac line by several degrees. This could throw off the timing for fading the lights/lamps.

A push button and a 3-way toggle switch were utilized to allow user override of the networks control of the lights.

CHAPTER 6

HIGH LEVEL SOFTWARE

The high level software resides in the PC. It is responsible for the creation of the control tapes. It accomplishes this by allowing task packets to be down loaded through the PC's serial port upon command. The program used at this time is MUSE.BAS. This program allows a list of command packets to be stepped through and down loaded one at a time through the serial port. The list of command packets can be altered simply by changing the DATA statements between lines 4120 and 4300. Knowledge of BASIC is not necessary to make these alterations to the program. It is very straight forward to alter the DATA statements to form a new list of command packets. The command packets should be listed in the order they will be down loaded to the presentation tape. If it is necessary to change the COM port of the PC, change all mention of COM1 to COM2 in MUSE.BAS.

MUSE.BAS can also be used as a development tool for expanding and debugging the system. MUSE.BAS was written for GWBASIC to be run on an IBM compatible. This program may need modification to run on other basic interpreters.

CHAPTER 7

TASK NODE SOFTWARE

The task node software is written in Intel 8051 assembly language. The source code for this project was assembled with the AVMAC development system created by AVOCET. Attempts at using another assembler will undoubtedly need modification. The AVMAC development system (AVOCET 1987) includes a two pass assembler, a linker, and the ability to write macros for library routines. The task node software consists of the Task Master operating system, the operating system global task list, and the application tasks.

7.1. Task Master Operating System

A common operating system resides on every task node. The function of the operating system is to receive properly addressed command packets from the RS-485 bus, and to act on them according to the present status of the system, and the status of the received packet's Pre-execution control character. The queued and synchronized tasks will always be queued up as long as the queue is not full. If the queue is full they are dumped. Queued tasks execute when they reach the top of the queue and the prior task has terminated. The same is true for the synchronized tasks, but they need a 'sync command' to begin. If the received packet is an immediate command it

will interrupt any queued or synchronized task. Control will return to the interrupted task when it terminates. If an immediate task is running it can not be interrupted by a received immediate task. The received immediate task will be dumped without responding in any way. The only command that can interrupt an immediate command is the reset command ('{%}'). The operating system also manages the task queue, but this should be transparent to the writer of an application task. What has to be known by a task writer is the state of the register space when control is passed to the task. Trashing dedicated registers risks crashing the operating system.

7.2. Task Node Internal Memory (256 Byte)

00H to 07H -- Queued task Register Bank 0 (RB0)

08H to 0FH -- Immediate task Register Bank 1 (RB1)

10H to 17H -- Unassigned by operating system Reg Bank 2 (RB2)

18H to 1FH -- Operating system Register Bank 3 (RB3)

20H to 22H -- Operating system status registers and pointers

23H to 25H -- Binary to ASCII conversion registers

26H to 4AH -- Open, Unassigned by operating system

50H to 5FH -- Host Command Packet (HCP) registers

60H & 61H -- Pointers to tail & head of task queue

62H to 68H -- ASCII buffer

69H to 6EH -- Timer registers

6FH to 7FH -- Open, Unassigned by operating system

80H to 9FH -- 8052 System Stack (Assigned by software)

0AH to FFH -- Task Queue (96 bytes)

Registers 00H to 0FH. Upon entering any task the Program Status Word (PSW) of the 8052 will be set to either RB0, if the task is queued, or RB1, if the task is immediate. This is done for the user by the operating system. For either register bank, queued or immediate, the contents will be the eight byte command packet.

COMMAND PACKET FORMAT

BYTE 1 POINTER TO THE ARGUMENTS

BYTE 2 TASK NUMBER

BYTE 3 HEX STATUS WORD

(bit #)

7 =0 PROCEED WHEN READY ":"

 =1 WAIT FOR PERMISSION "?"

6 =0 DO ONCE "."

 =1 RE-QUEUE "+"

54 =00 NOT YET BEGUN

 =01 BEGUN

 =10 WAITING FOR "\$" SYNC SIGNAL

 =11 FINISHED

3210 = NUMBER OF ARGUMENTS (0 - 5)

BYTE 4 HEX ARG1
BYTE 5 HEX ARG2
BYTE 6 HEX ARG3
BYTE 7 HEX ARG4
BYTE 8 HEX ARG5

At the start of every task R0 points to the argument space and R3's lower 3 bits hold the number of arguments. Up to five bytes of data can be passed in BYTE 3 to BYTE 7. When writing a task this data, combined with the unassigned register space, forms the boundary conditions for the task implementation.

Registers 10H to 17H, 26H to 4AH, and 6F to 7F are not used by the operating system and are available for application tasks.

Registers 18H to 22H are used by the operating system and should not be altered by the task writer.

Registers 23H to 25H are used by routines 'bin_to_asc' and 'bin_to_bcd' to convert binary numbers to BCD or ASCII.

Registers 50H to 5FH is the Host Command Packet area, used by the operating system to create the Command Packet.

Registers 60H to 61H are 'POPQPTR' and 'POPQPTR', the task queue pointers.

Registers 62H to 68H are used by as_hex to convert ASCII code to hexadecimal (See subroutine as_hex). With the pair of subroutines

'bin_to_asc' and 'as_hex' integer data can be easily passed back and forth from the network.

Registers 69H to 6EH are timer registers that, along with timer0 can support a real time clock. For the museum project the timer0 interrupt was utilized for other purposes and the timer routine was deleted. If a real time clock is necessary the routine will have to be rewritten or copied from an older version of the Taskmaster operating system.

Registers 80H to 9FH are the 8052 system stack. It contains 32 bytes which might be much greater than user needs. If this is the case the system stack can be shortened.

Registers A0H to FFH are the 96 byte task queue. If the queue can be shortened, as in the museum project, register space for application tasks can be utilized. If a larger task queue is necessary the external RAM version of the Taskmaster should be used.

All changes to the assigned register space should be fully documented.

The application programs created for the museum were, lite##.asm (Where the ## is the address of the task node) and Projectr.asm. Lite##.asm uses internal registers space 2FH to 4AH, and Projectr.asm uses registers space 2AH to 48H.

CHAPTER 8

TASK SOFTWARE IMPLEMENTATION

From the view of the node, a task is simply a subroutine which resides within the system ROM. These tasks are initiated by the operating system based on the received command packets. There are two types of tasks, global tasks and application specific tasks. Global tasks are common to all nodes. They provide basic operations for the system. Application specific tasks provide specialized operations based on the specific hardware and function of each individual task node.

8.1. Global Task

The purpose of the Global Task Library is to provide a set of basic utilities for input/output operations and system development. The following is the list of tasks in the global library.

TASK0 -- Reset the task node

TASK1 -- Clear the task queue

TASK2 -- Abort the current task

TASK3 -- Start a synchronized task on the queue

TASK4 -- Pause the system

TASK5 -- Resume system operation

TASK6 -- Set time out register

TASK7 -- Null task

TASK8 -- Task with infinite loop

TASK9 -- Display task queue to the host

TASKA -- Send task node address to the host

TASKB -- Send to task data memory transfer

TASKC -- Data memory to host transfer

TASKD -- Host to external memory transfer

TASKE -- External memory to host transfer

TASKF -- Sensitize local asynchronous port

TASK10 -- De-Sensitize the asynchronous port

TASK11 -- Sensitize the parallel port

TASK12 -- De-Sensitize the parallel port

TASK13 -- Set time delay

TASK14 -- Audio signal

TASK15 -- Friendly blink of LED

8.2. Application Specific Tasks

For the museum project there were two basic functions, control the slide projectors and control the lights. The implementation of each function was then broken up into libraries of tasks. The small board computer purchased from Binary Technology was assigned the projector control tasks and was arbitrarily assigned the address 01. The light control tasks were assigned to six boards designed as minimum component nodes. They were assigned the addresses 06 to 0B.

To construct a packet for the light controller it is first necessary to map each of the lights to be controlled with a task node and associated light driver. Each node has light drivers numbered 1 to 4. As depicted in figure 8.1 there are twelve overhead lights that need to be controlled. Each was assigned the node address and driver number as shown. These assignments are the reference between the lights to be controlled and the node address\task argument when creating a task packet (see Appendix A.2 task51).

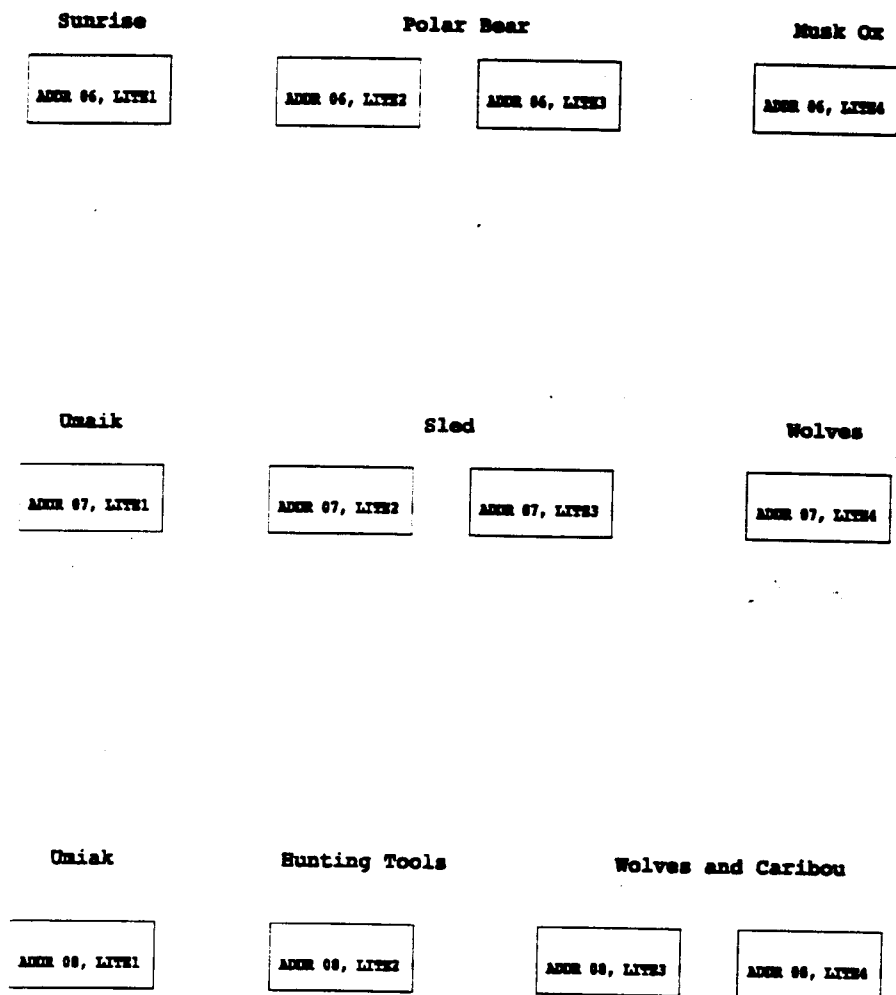


FIGURE 8.1: MUSEUM LIGHTING ASSIGNMENTS

To create a packet for the projectors it is only necessary to know the defined projector number. For this project the projector to the left is defined as number 1. The application specific task library for the slide projector control task node is a collection of tasks that provide the system scheduler, the tape recorder in this case, the capability to control one or more slide projectors remotely. Special effects such as superimpose, and variable fade-in and fade-out rates are accomplished by using this task library. Besides tasks for creating the special effects, the library also includes other utility tasks for setting up and monitoring the operation of the projectors. The following is a list of tasks available in this library [7].

TASK40 -- Initialize projector connections

TASK41 -- Fade lamp (s) in / out

TASK42 -- Display lamp status (on / off)

TASK43 -- Set lamp / frame display option

TASK44 -- Set tray size

TASK45 -- Home tray

TASK46 -- Advance slide(s)

TASK47 -- Initialize frame counter(s)

TASK48 -- Display frame counter(s)

TASK49 -- Delay time

TASK4a -- Raise the projector screen

TASK4b -- Lower the projector screen

The application tasks for the light controller follow,

TASK50 -- Delay the execution of task40 by xx secs

TASK51 -- Fade 4 lights to varying levels in xx secs

8.3. Algorithm For Light Control

Software for most of the above tasks are very straight forward, but the algorithm for fading a light (or a projector lamp) requires some background information as an aid in understanding. A lamp can be modeled as a pure resistive circuit. Under this condition the following is true:

$$p = iv(8-1)$$

Where,

p = the instantaneous power

$$v = V_m \cos(\omega t)(8-2)$$

$$i = I_m \cos(\omega t + \Theta)(8-3)$$

$\Theta = 0$ for a pure resistive circuit

Since,

$$i = V_m \cos(\omega t)/R(8-4)$$

Where, R is the resistance of the filament.

Then,

$$p = [(V_m)/R] \cos(\omega t)(8-5)$$

and the average power for a half cycle of the 60Hz input is then,

$$P_{av} = \frac{V_m}{T R} \int_{t_s}^T \cos(\omega t) dt \quad (8-6)$$

Where, P_{av} = the average power

T = the period for a half cycle

t_s = starting time for the sinusoid $0 < t_s < T$

While the resistance of the filament, and the light energy emitted, are extremely non-linear with applied voltage it is still true that the lights intensity rises and falls in relation to the area enclosed by the square of the sinusoid $\cos(\omega t)$. The concept is best understood if viewed graphically. The area under the sinusoid can be viewed as a Voltage Time Area (VTA). As shown in Figure 8.1, if $t_s = 0$ then the maximum power is being dissipated, and the VTA is the entire area under curve from 0 to T . As t_s is increased the area under the curve is decreased until at $t_s = T$ the $VTA = 0$ and the average power, and light energy dissipated, is also 0. If the VTA can be controlled the light intensity, at any moment in time, is also controlled. Triacs and the SSR function in just this way. Enable them during a VTA and they conduct until the zero-crossing occurs ($\cos(\omega t) = 0$).

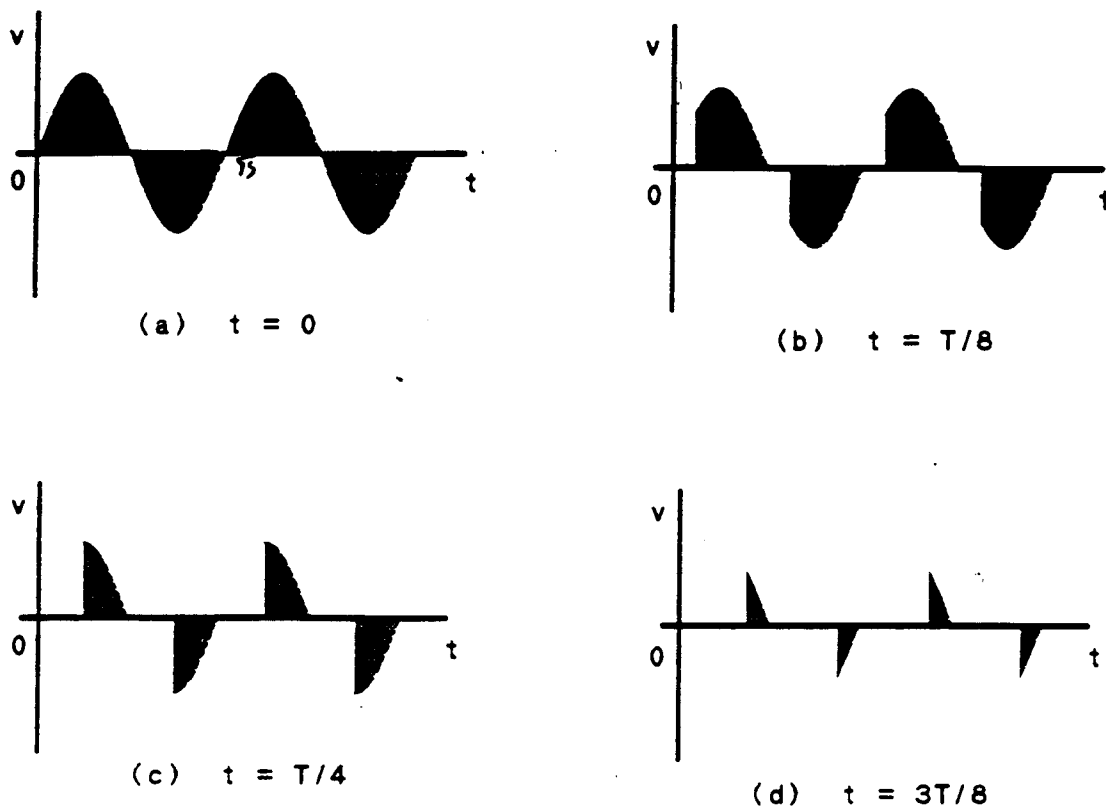


FIGURE 8.2: VOLTAGE TIME AREA POWER ANALYSIS

Since the microcontroller is, by definition, a digital device t_s can only be defined at discrete time intervals. By dividing the wave into sections of time it is possible to abandon the notion of t_s and refer to a section number as defining the VTA. Since the processor has a set processing speed, and the triacs and SSRs have minimum gate enable periods, an upper bound exists for the number of 'sections' that the 120Hz wave can be divided. For the light control nodes this was found to be 120 sections, for the projector control node it was 30. The principle reason for the difference in sections is the time interval that the gate has to be pulsed to initiate conduction. For the triacs this pulse was 10 μsec and for the SSRs it was on the order of 100 μsec . While it is advantageous to have more sections, in terms of smooth intensity transitions, the human eye has limits. For the projector, where transitions from full off to full on, or visa-versa, are less than 6 seconds, the 'stepping' of the sections goes unnoticed. For the lights the stepping is unnoticed below a fade time of about 12 seconds for full off to full on. The sections were referenced to the 120Hz signal with section 0 corresponding to the start of the wave.

Along with controlling the intensity of the lamps it is necessary to control the rate of change in intensity, defined here as the fade rate. To control the fade rate it is necessary to control the number of half cycles that the section number remains constant before incrementing (fade down) or decrementing (ramp up) the section number. This number of half cycles is

defined as RATE. The formula for RATE is,

$$\text{RATE} = (120\text{Hz}/(\#\text{sections}/\text{half-wave}) * (\#\text{seconds}) \quad (8.7)$$

For example a half wave occurs 120 times a second (period = 8.33ms).

If it is desired to fade the light from full off to full on in 5 seconds,

Rate = (120/120 * 5). It is necessary to hold the section number constant for 5 half cycles before decrementing them. (Initially the section number would be 120, full off, its final value will be 0, full on.) For the projector, RATE = (120Hz/30sections/hw) * 5seconds = 20 sections. To achieve a 5 second fade for the projector the section number would have to remain constant for 20 sections before it is decremented.

8.4. Pseudocode For the Fade Algorithm

Each projector task node has the ability to control four projectors and each light control node can control four lights. It is assumed that information about the fade rates and final values are available in the data fields (See section 3.2).

The fade algorithm used for the lights and projector lamps are slightly different in nature. The projector lamps can only have full on or full off as final values. The lights can have final intensities from 00% to 100% in 10% increments. Thus final values for the lights of 10%, 20%, 30%, etc. are possible. While this does make the software to fade the lights more complicated, then that of the lamps, the core of both fade routines is the following algorithm.

Pseudocode to fade a projector lamp/light:

Get the final value from data field.

Translate and store final section # for each lamp in FVAL

Get the rate value from data field.

Translate, store rate in array RATE and WKRATE for all lamps.

(WKRATE is the working copy of the RATE value.)

Zero Crossing Loop:

Wait for the zero-crossing.

Allow about 7% of half wave to pass before counting sections

Clear current section counter

Start section timer, timer set for one section.

Half Cycle loop: loop repeated 120 times per second.

Compare PVAL(i) to section number, equal? fire the triac/relay

Wait for section timer to overflow signifying end of section

If all sections tested (120 for lights or 30 for projector),

then half cycle loop completed, exit half cycle loop

else increment section number and repeat half cycle loop.

Reset i = 0

Update loop: Update WKRATE(i) and PVAL(i)

If WKRATE(i) != 0 then decrement it

else reload WKRATE(i) with RATE(i) and

if PVAL(i) < FVAL(i) increment PVAL(i)

else if PVAL(i) > FVAL(i) decrement PVAL(i)

else PVAL(i) = FVAL(i),set DONE flag for lamp(i)

If (i) = 3 go on as all lamps tested,

else increment (i) and repeat Update loop

If DONE flag set for all lamp(i) then task done, exit

else go to Zero Crossing loop

Explanation is necessary for the 7% delay before beginning the section count. The zero-crossing detector may be buffered by a transformer whose output may lag the actual 60Hz line voltage by a few degrees. Since the VTA increases slowly during the beginning of the half wave, little is lost in terms of light intensity by delaying, but synchronization of the node with the 60Hz input is guaranteed. At the end of a half cycle a similar situation exists. If the triac/relay is fired within the last 15% of the half cycle the VTA will not allow the lamps to radiate appreciable light. This period is used to execute the necessary instructions to allow the processor to control the lamps. The timer for the sections should be set based on these constraints. Upon reset of the 8052 all lamps are turned off, and the arrays PVAL and FVAL are initialized to the maximum section number (120/30), equating to full off for all lamps.

The other tasks are fairly routine and should be easily understood (see appendix A.2)

CHAPTER 9

PRESENTATION TAPES

The creation of a taped presentations is a two step operation. The audio track must be produced on the left channel and the task list must be down loaded to the right channel. A list of commands that will perform the desired tasks, in the desired order, must be created and interfaced with the high level software (MUSE.BAS mentioned above). The audio portion of the tape is played and the list of tasks are stepped through and down loaded at prescribed moments.

9.1. Presentation Tape Creation Equipment

The three pieces of equipment necessary to create the presentation tapes are, a PC, a modem, and a multi-track recorder.

The multi-track recorder used for this project was the MR-200MK II. The MR-200MK II can perform the basic multi-track recording, overdubbing, ping-pong recording and remixing necessary to create the presentation tapes. While the following information might satisfy your needs, the MR-200MK has functions beyond those mentioned here. The manufacturers instruction manual should be read before attempting to make a tape.

The PC is connected to the modem through the serial port. All communications are RS-232 so the NIUs are not needed. The output of the

modem is connected to the line2 RCA jack of the MR-200MK II 'Line in' jack panel. The input to the modems RCA jack is connected to the track2 RCA jack of the 'Tape Out' jack panel. Use RCA plugs to connect the tape players output to the MR-200's 'Line in' jack panel. The left channel should be connected to line1 and the right channel is connected to line3 of the panel.

To up-load, to the tape player, the modem should be set on F (full duplex) and A (answer mode). To monitor the results after the tape is created, switch the modem from A (answer) to O (originate) mode, and play the tape.

9.2. Creating Taped Presentations

The audio portion of the tape has to be recorded first. This can be done in several ways depending on the number of tracks to be overlaid on the single channel available. Only two independent tracks can be recorded at a time, but through 'ping-ponging' any number can eventually be overlaid on the same track.

Head phones are very useful for this task and allow monitoring of the volume level while recording. The MR-200 allows inputs from Microphones and all standard musical electronic equipment, but it is assumed that the sounds to be recorded are on cassette tape and a standard tape player is available.

For example, suppose that a tape is to be made with narration, music and animal sounds on the defined audio track1 (left channel) with the control signals on track2 (right channel).

The MR-200 MK can only record on two of the four tracks at the same time. Choices are limited to tracks 1 or 3 and tracks 2 or 4. The narration should be used as the reference and the music and animal sounds should be synchronized to it.

First decide where the music and animal sounds should go in reference to the narration. Give labels to these bits of music and animal noises and reference these labels to their starting points on the tape they are recorded on. This is done with the tape counter, reference the start of the tape to 000 and write down the count at the start of the bits desired. This list and the tape should be stored together for future reference.

Start by recording the narration on track 1.

Set the meter switch to PGM.

The PAN control should be turned fully to the left if a channel is to be recorded on track 1 or track 3 and fully to the right for track 2 or track 4. For this case turn the PAN control fully to the left for both line1 and line3 since the channels are being recorded on track 1.

Play the tape to be recorded and adjust the level control for the PGM L meter. Set the control so the 0db LED blinks often.

The output can be monitored with the headphones. The volume of the headphones can be raised with the PHONES control. The volume level of the headphones has no effect on the recording level.

Set the PITCH control knob to mid range.

Set the REC FUNCTION switch to track 1, the stand-by warning LED

should flash.

Press the REC button and the the MR-200 will start recording.

When the tape is finished, turn off the REC FUNCTION switch and replay the tape to make sure what you have is what you want. Repeat process until the tape is satisfactory.

Reset the tape counter on the MR-200 to 000. Play the tape. When the narration reaches a spot where music, or animal sounds, should be overlaid, write down the counter value, and the associated label for the particular music or animal 'bit'. Set up the tape player so that the first 'bit' to be overlaid is the next thing to be played.

Since the 'bits' are going on track 2 turn the PAN control fully to the right for both line1 and line3.

Adjust the level control for the PGM R meter.

Use the headphones to monitor the timing of the overlay.

Set the PITCH control knob to mid range.

Set the REC FUNCTION switch to TRACK2.

Reset the counter to 000.

Press the REC button and the the MR-200 will start playing track 1 and recording on track 2.

When the counter equals the label address, start the tape player. By using the level control the 'bit' can be faded in.

When the 'bit' is over, stop the tape recorder, or fade it out with the level control.

Pause the MR-200 to set the next 'bit' up for overlaying.

Repeat the above procedure until all sounds are overlaid.

Turn off the REC FUNCTION and rewind and replay the tape to make sure it is recorded correctly. It is now necessary to overlay track 2 on track 1.

Rewind the tape.

Switch the PAN control for line1 and line2 fully to the left.

Set the REC FUNCTION switch to track1

Press the RECORD button. Stop the tape when it is over.

The audio portion of the tape should now be finished, and the control portion can be added.

At the beginning of each presentation the reset command should be included. This does two things: It resets the task nodes to a known state, and each node executes a Friendly Blink (FB). This Friendly Blink shows that the RS-485 communications is working and that the nodes are operating. If one of the task nodes does not blink its LED then that node needs to be checked.

It is assumed that the MUSE.BAS file has had its DATA statements changed to include the list of command tasks for this particular recording. Make sure each basic file created has a unique name that easily identifies it with it's presentation tape. The basic file to down load the tasks will be referred to as FILE.BAS. The task commands in the DATA statements have to be the right commands, in the right order, or the network probably will not function correctly. Check them carefully.

Run GWBASIC and load and run the FILE.BAS program.

Rewind the presentation tape and make another list of counter numbers, based on when the tasks are to be down loaded. Hopefully they do not need labels, as the prepared DATA statements should be in order, and will be 'entered' one after another.

The modem should be set on F (full) and A (answer).

Slide the REC Function switch to track2.

Turn the PAN switch fully to the right for line2 and the PAN switch for line1 fully to the left.

With the headphones the sound of the FSK 2000Hz signal should be present. If the serial port is exercised a chirping, as the frequency is shifted, should be heard in the right headphone.

With the FILE.BAS program set to the first entry, press the record button. When the counter numbers match those of the command file list, down load the task (the enter key) and advance to the next task packet (the space bar). Continue until the tape is finished.

If an error is made, or to check the results, exit the FILE.BAS program and execute PROCOMM or some other terminal emulation program. The default communications parameters are, 300baud, 8 data bits, no parity, 1 stop bits.

Switch the modem to O (originate) and play the tape. If the task packets are not being received, or are out of sync' with the tape, either rerecord the entire track 2, or possibly 'patch' in the correct commands.

Patching in commands is possible but can be frustrating. This entails going back and forth from PROCOMM and the FILE.BAS program, switching the Answer and Originate toggle on the modem, and switching the record and play options on the MK-200. If unwanted signals are recorded on track 2 it is not important as long as the task packets are uncorrupted. The operating system will reject the unwanted signals. This process can be difficult, so concentrate and have patience.

CHAPTER 10

OPERATION OF PRESENTATION SYSTEM

The operation of the system consists of turning on the power switch, starting tape deck #1 and letting it run to the completion of the presentation. At the end of the presentation the tape should be rewound and the power switch turned off. If the presentation is not allowed to run to completion it may be necessary to manually reset the projector trays to 00, raise the projector screen, and bring up the houselights under manual control. The system at this point is reset and should run correctly as outlined above. NOTE! The lights must be released from manual control before running a presentation. The manual control overrides all commands to the task nodes, preventing the execution of received tasks.

10.1 Operational Components

The power switch is ganged to power up all necessary devices to run the presentation with the exception of the light control box and the mixer control panel. The mixer has to be on a separate circuit because it also controls the output of the VCR. The light control box is always on and is not switched, this is necessitated by the need for manual (emergency) control over the lights. The devices controlled by the power switch are the projectors (#1,#2), projector control box power supply, 60Hz input to the

projector control box, modem power supply, tape player #1, and the amplifier for tape player #1. If any of these devices are not powered up the system will not function properly.

10.2. Operation of the Light Controller.

The light control box has 2 functions.

1. Provide computer control over the lights while a presentation is running.

2. Provide manual control of the lights at all times.

Computer control of the lights is done by the taped presentation and if operating properly, requires no user involvement.

Manual control is activated by the square red button and the toggle switch on the left side of the light control box. To engage manual control, push in the red button and the lights will come up to 50% intensity level. The 3 position toggle, to the right of the red button, allows the lights to be faded up or down (up is up, down is down, and the middle position is no change.). The red LED under the switch indicates that the manual control is in operation. When manual control is initiated, it saves the current state of the lights. When the manual control is released (The red button is out and LED is off) the lights return to the state they were in before manual control was initiated. The task nodes, that are affected by the manual control can be selected by the switch panel located under the middle section of the front of the box, between the manual control and the friendly blink

LEDs. These 10 switches correspond to the maximum of 10 nodes that could be installed in the control box. If set, the lights assigned to the node will be controlled by the manual override. If not set the manual override will be ignored by the particular node.

10.3. Operation of the Projector Controller

The projector controller, if operating properly, should require no user involvement. At the start of every presentation the trays should be set to 00 and the projectors power switches should be set to 'Fan'.

If the presentation is shortened, if the slide show is not presented, it is only necessary to rewind the tape and reset the projector trays. The system should be reset at this point and ready for the next presentation. If the presentation is run to completion the trays should be reset automatically.

CHAPTER 11

MAINTENANCE

The system is fairly complex and when not operating properly debugging can be involved. Intermittent problems are the worst and require patience and perseverance.

The necessary tools for debugging are clip leads, push type and normal, VOM, 2 channel Oscilloscope, a terminal or computer (that can emulate a terminal), and connectors for the serial communication.

A soldering setup and associated equipment are also required (wire, IC's, screwdrivers, etc.) for most repairs.

Most problems encountered have been attributed to faulty connections and ground loops. To prevent ground loops all power supplies grounds have been referenced to earth ground. Connection problems often result from cold solder joints, oxidation of connectors, or faulty crimp connections. The following problems may be encountered.

11.1. Tasks not Executed by any Nodes.

Possible causes:

- A. Tape player down
- B. modem down
- C. NUI down

D. Task node 75176 transceiver is loading the bus

1. With the oscilloscope check the output of the NIU to the network bus (red and green wire connections) when packets should be present on the line. If none of the nodes are executing tasks there should be no signals, or garbled signals, present. If valid signals (0 to 5volt transitions) are present, and tasks are still not being executed, then it is most likely a connection problem with the transmission lines supporting the network.

2. If the signals are not valid, disconnect the NIU from the network and check the signals. If the signals are now valid either one (or more) of the 75176 transceivers is loading the bus or there maybe a ground loop, caused by one of the power supplies losing its reference to earth ground. If the signals are still not valid, disconnect the NIU from the modem and check the modem output. If valid RS-232 signals are present the problem is either the connections between the modem and the NIU or the NIU itself. If RS-232 signals are not present, when they should be, check the modem switch settings F (full),O (originate) and power connections.

3. If signals still are not valid, disconnect the modem and check the output of the tape player for valid FSK signals. If the signals are valid the problem is the modem. If signals are not valid the problem is the tape deck or the actual tape. Hopefully replacement parts will be obtained for all these low cost components. If they exist replace those that appear to be defective. If, after replacement of the part(s), the problem ends, attempt to restock the item as it will likely need to be replaced again later. If

the problem persists, persevere.

If the problem is a faulty NIU, chips can be switched with the offending unit and the replacement unit until the malfunctioning chip is found and replaced.

11.2. Communications OK but Lights/Lamps Fade Incorrectly.

Possible solutions:

A. Zero cross detector is not functioning correctly.

B. Zero cross detector is functioning but it's input is not synchronized with the PPL 60Hz signal.

C. RS485 communication is being compromised.

D. Devices are not connected/functioning properly.

A. The zero cross detector feeds different pins on the 8052 for the lights and the projector controllers. The following instructions are valid, but need to be check independently for each of the above. For the light controller check the terminal marked zero cross,near the front of the box, for a clean 60Hz square wave. For the projector controller check the pin #12 (IE0) of the 8052 for the same.

If the zero cross is not present check for connection problems. Look at the input to the zero cross detector, the output of the LM311, and the output of the opto isolator. Replace chips as necessary. If the power supply has a bad, or worse intermittent connection, it is possible that the

number of zero crosses are not being counted correctly. I.E. if -5volts is out only 60 transitions per second will be counted.

B. Check the zero cross for synchronization with the 60Hz input from PPL (120volt AC).

C. It is possible that characters are being missed and not all of the commands are being received. This can be checked by connecting a PC, through a NIU to the RS-485 network (Make sure the transmitter is disabled to keep it 'off' the bus.). Monitor the communication and see if packets are being received correctly.

D. If a part is faulty try to isolate the problem. Make sure the communications are working. Either monitor the network, or directly send commands from the PC. Reference one channel of the oscilloscope to the square wave output of the zero-crossing detector. With the other channel of the oscilloscope monitor the relevant pins of the 8052 that drive the opto-isolators or the projector solid state relays. Check if the pins are being driven correctly for the received task. Then check the associated output pins of the 7407, and the inputs to the 3010 (If 3011 ICs are present they are functionally identical to the 3010.). Replace ICs as necessary.

11.3. Lights Stay On/Off all the Time.

If the zero-crossing circuit fails none of the lights associated with the node will function properly.

If only one light is on/off it is most likely a triac that is

defective. For quick replacement Radio shack sells a 400volt 6 amp Triac.

If the light stay on all the time, disconnect the light control box from the 120 volt supply. Remove the 3010 opto-isolator for the suspected triac. Reconnect the power. If the light stays on it is undoubtedly the triac and it needs to be replaced. If the light stays off send a command to the node to turn on the light under test. Check the output of the 8052 port pin driving the light. If a pulse is not observed, the communication is down, or the 8052 is defective.

If the proper pulse is present check pin 1 of the 3010 referenced to the zero-cross. If this is OK, power down the circuit and check for proper continuity according to the schematic. If this all checks out the triac should definitely be replaced.

11.4. Projector Lamps Stay On\Off all the Time

If the zero-crossing circuit fails neither of the projector lamps will function properly.

Reset the system. Make sure the projectors are set on 'Fan'. The problem is probably an open connection. The SSRs are industrial grade and are rated at four times the amperage required by the lamps. To check, down load a fade command to the projector in question. If the lamp is on all the time try to turn it off, and visa-versa. Watch the input pin to the SSR and try to observe the expected pulse. If the pulse is present, and the lamp does not respond, it is a faulty SSR or, more likely a bad connection. If the pulse does not show up, check the 8052 port pin that drives the SSR.

CHAPTER 12

CONCLUSIONS AND RECOMMENDATIONS

12.1. Conclusions

The objectives of the multi-media presentation system based on a distributed control network were met. In July of 1991 the network was installed and one month later is still operating as designed. Modifications to the network were:

- 1) The operating system was altered, principally by moving the task queue from the external RAM to the internal 8052 RAM space. This improved the speed of the operating system and reduced the chip count by two.
- 2) The chip count for a task node was reduced from 17 chips to 5. This gain, even for a modest network, would be substantial in terms of cost and board space.
- 3) The projector screen controller and light controller were implemented and worked as designed.
- 4) Modifications to COLAN V task library and hardware were successful. All COLAN V application tasks are operational. Subroutines `as_to_bin` and `bin_to_as` were added to allow binary and ASCII bytes to be easily translated. This allowed math to be done in binary and easy translation from binary for serial communication.
- 5) The network was simple enough for the museum staff to feel

comfortable with its operation.

6) The recording of audio programs on one track of the tape and the control signals on the other was proven to be an excellent method of storing multi-media presentations.

This project successfully showed that distributed control networks can be used to control a multi-media presentation system. The addition of nodes as modules allows easy expansion of the network, while maintaining low costs.

12.2. Recommendations for Future Research

While the multi-media presentation system is successful there are many things left to do to make it a truly functional product. Also several improvements could be made in relation to the general format of the network.

- * To make the multi-media presentation system truly a flexible and useful product it needs a more powerful user interface. The MUSE.BAS program while functional is awkward. The necessity of using Data statements is tedious. It would be helpful if, during tape creation, the program offered the ability to back up through the task list when a mistake is made. Also there should be a shell to allow terminal emulation without exiting the program.

- * Back up components should be obtained for all critical parts.

- * Add a task node to control the VCR. Tasks could be written for the light controller that would close the power switch, rewind the tape, open the

power switch, and turn the tape player off. This would reduce the user involvement to turning on the tape and starting the presentation.

- * The tape player restricts the baud rate by requiring FSK encoding of the data. This restriction still should allow a baud rate of 600 bps, doubling the communications bandwidth. Also the nodes could have automatic baud rate synchronization similar to that of MDP-51.

- * Move the projector screen controller over to node 01 where it belongs.

- * The light controller nodes now drive four lights, they could drive eight. This would be ideal for updating lights as one byte. This would reduce cost dramatically.

- * Run the zero-crossing as an interrupt. This would eliminate the need for task51 being restricted to only being a queued task, and allow for a more flexible system.

- * The fifth argument used to trigger a delay in task51 is unnecessary and could be replaced by just setting a dedicated flag in task50.

- * For even a modest network it can be costly if the microcontroller boards have to be bought. A node can function with as little as five chips. A circuit board should be designed specifically for the network. It should be designed to interface with the DDT debugger easing system development.

- * Error checking could be added. A simple checksum, could be implemented with the loss of only one dedicated register and 2 instruction cycles per input byte.

- * The PC should control one of the DB-25 pins on the serial port to disable\enable the RS-485 driver. This would allow the PC to monitor the bus from remote locations without the need to worry about it loading the RS-485 bus. The modem NIU to the tape player should also have its 74123 used to disable its RS-485 driver when not in use (Now it is always on the bus).

- * The CSMA protocol should have the collision detect (CSMA\CD) added to it. This possibly could be done by having the transmitting node listen to the first byte transmitted, since the communication is full duplex. This would necessitate changing the algorithm for disabling the bus drivers in the operating system.

- * As the operating system is now there is no convenient way to terminate an immediate task. There should be a termination task, or possibly let a received immediate task terminate a running one.

- * Add more arguments to the task list. Possibly have dedicated register space used for both immediate and queued tasks, and just flag the type of task. Also there is no need to pass the task the task number, this would easily free up one byte.

- * Replace the 75175/75174 on the NIUs with the 75176

CHAPTER 13

BIBLIOGRAPHY

- 1.) Appleby D., Thomas C., Shaffer J. Cost-effective Control for Small Batch Operations. I&CS, November 1989.
- 2.) Herzog, J. H. and Zhang, T. G. "A Design Methodology for Distributed Microprocessors in Real-Time Control Applications." Proceedings, Second International Conference on Computers and Applications, Beijing China, June 1987.
- 3.) Horowitz P., Hill W. The Art of Electronics. Cambridge University Press, Cambridge 1987.
- 4.) Intel Corporation. Microcontroller Handbook. 1989.
- 5.) National Semiconductor Master Selection Guide. 1989.
- 6.) Nisley E. "A Network for Distributed Control." Circuit Cellar Ink., Aug\Sept,Oct\Nov 1989.
- 7.) So, D. Multi-Media Presentation System For Control Oriented Local Area Networks (COLAN). Unpublished Masters Thesis, Oregon State University, Corvallis, Oregon, 1990

- 8.) Tanenbaum, A. Structured Computer Organization.
Englewood Cliffs, NJ, Prentice Hall, 1988.

APPENDIX

APPENDIX A

A.1. Specifications of Museum Project

The projector specifications were:

Advance/reverse the slide tray (1.5 amps max)

Fade the slides on and off at programmable fade rates
(2.5 amps maximum)

Home the slide tray

Lower/raise the projector screen (5.5 amps maximum)

The light specifications were:

Control the fade rate and intensity of the lights.

(presently 150Watt @120V)

Allow manual override control of the system

A.2. Application Specific Task Library

The following is a list of the tasks in the Application Specific Task Library for the museum project task node. The notations that are used in the task definitions are:

Task Command Packet Format -- {AA P NN S DD}

AA : Destination Node Address

P : Pre-Execution Control Character

: Queued Task
 ? Synchronized Task
 ! Immediate Task

NN : Task Number

S : Post-Execution Control Character

. Execute Once
 + Re-Queued Task

DD : Data Field

AB Argument #1
 CD Argument #2
 EF Argument #3
 GH Argument #4
 IJ Argument #5

Note that due to overlap in I/O ports by the projector tasks and the light controller tasks the task libraries for each function should not be included in the same ROM.

For a list of application specific tasks see section 8.2.

Projector Specific Tasks

Task Number:40

Function:Initialize projector connections

Format:{01:40.0B}

Description: This task provides information to the task node specifying which projectors are connected to the system. System default is projector 1 & 2 being connected.

Arguments: The hexadecimal value of argument B indicates which projector port is connected.

Examples:

Data Field	Projectors Connection			
	4	3	2	1
B				
1				X
2			X	
3			X	
4		X		
5		X		
:	:	:	:	:
:	:	:	:	:
C	X	X		
D	X	X		X
E	X	X	X	
F				

Task Number:41

Function:Control fade rates and intensities of the lamps

Format:{01:41.ABCDEFGH}

Description: This task is used to specify the fade rate of each lamp and the final intensity level. The fade rate is the time it takes to reach the final intensity level.

Arguments: The argument numbers correspond directly to the projector numbers. Data fields A, C, E, and G specify the fade rates and data fields B, D, F, and H specify the final intensity levels for lights 1 to 4 respectively. The following lists the possible fade rates and intensity values:

Fade Rate Table (Arguments A, C, E, G)

Fade Value	Fade Rate (secs)
0	Instantly
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	12
C	15
D	20
E	25
F	30

Final Intensity Level (Args B,D,F,H)

Intensity Value	Intensity Level
0	No Change
1	ON
2 - F	OFF

Examples: {01!41.314F} Immediate task, fade p1 to full on in 3 secs. and fade p2 to full off in 4 secs.
 {01:41.00630741} Queued task, no change to p1, p2 full off in 6 secs., p3 immediate full off, p4 full on in 4 secs.

Task Number:42

Function:Display lamp status

Format:{01:42.}

Description: This task returns the ON / OFF status of the projector lamps to the host computer.

Arguments: None.

Examples: Format for status message returned to Host:

Lamp Status [ON:OFF:OFF:ON]

The projector numbers corresponding to the lamp status are inferred by the positions in packet,
[Projector 1:Projector 2:Projector 3:Projector 4].

Task Number:43

Function:Sets lamp status / frame counter display options.

Format:{01:43.AB}

Description: This task sets and clears the flags for determining whether to return the lamp status packet and the frame counter packet after the execution of each task.

Arguments: Argument A sets the frame counter flag (lamps.w.6) and argument B sets lamp status flag (lamps.w.7). The arguments have two options: any value other than 0 sets the flag, and a value of 0 clears the flag. Once the flag is set, the corresponding status/counter packet will be return upon the completion of each task. The format of the return packets is the same as those from executing task 42 and task 48.

Data Field	Packet(s) Returned
A = 0, B = 0	None
A = 0, B = 1 - F	Lamp status packet
A = 1 - F, B = 0	Frame counter packet
A = 1-F, B = 1-F	Lamp & Frame counter packets

Task Number:44

Function:Sets the slide projector tray size

Format:{01:44.ABCD}

Description: This task sets the tray size on each of the slide projectors connected to the task node.

Two possible tray sizes are: 80 frames and 140 frames per tray. The system default is 80.

Arguments: The arguments are used to set the size of the tray on each projector: the value 0 is for 80 frames per tray and any nonzero value is for 140 frames per tray. 1 is arbitrarily used in the following table.

Projector Tray Size Table

Data Field	Tray Size			
ABCD	Proj#1	Proj#2	Proj#3	Proj#4
0000	80	80	80	80
0001	80	80	80	140
0010	80	80	140	80
0011	80	80	140	140
0100	80	140	80	80
:	:	:	:	:
:	:	:	:	:
1100	140	140	80	80
1101	140	140	80	140
1110	140	140	140	80
1111	140	140	140	140

Task Number:45

Function:Home Tray

Format:{01:45.0B}

Description: This task rotates the tray back to its original starting position. The tray will be automatically rotated in the shortest direction, either clock-wise or counter clock-wise.

Arguments: Argument B is used to specify which tray to home. The tray number corresponds to the projector number. Arguments greater than 4 will error out with no effect.

Data Field	Tray #
0	No Operation
1	Home Tray #1
2	Home Tray #2
3	Home Tray #3
4	Home Tray #4

Task Number:46

Function:Advance tray forward or reverse

Format:{01:46.AB}

Description: This task causes the projector tray to advance either forward or reverse depending on AB.

Up to four trays can be advanced at once.

Arguments: The bit pairs in Argument A are encoded to represent the command to step the tray forward or reverse. Alternatively the arguments AB could be thought of as one hex byte.

Projector Bit Encoding

p4	p3	p2	p1
r/f	r/f	r/f	r/f
b7/b6	b5/b4	b3/b2	b1/b0

P_i = projector(i)

r/f = reverse bit or forward bit

b(i) = the bit position of the r/f flag

If a bit for one of the projectors is set then that projector will be stepped (if it is connected). If both projector bits are set the

default is to step the tray forward. To create the arguments, form the binary word from the bit patterns given above and translate this word into hexadecimal.

The other approach is to start with the argument AB as being the summation of hexadecimal numbers that reference the appropriate commands as hex numbers. The method is to add up the hexadecimal values for the desired arguments.

Hex Encoding for Tray Direction Arguments AB.

Proj #	Tray Direction	Hex Arg.
Proj #1	Forward	01H
	Reverse	02H
Proj #2	Forward	04H
	Reverse	08H
Proj #3	Forward	10H
	Reverse	20H
Proj #4	Forward	40H
	Reverse	80H

Example: To: Step p1 forward, p2 backward, p3 nop, p4 step forward

p1 forward => 01H

p2 backward => 08H

p3 nop => 00H

p4 forward => + 40H

AB = 49H

Therefore the task packet would be {01!46.49}
for an immediate task or {01:46.49} for a queued
task

Task Number:47

Function:Initialize frame counter

Format:{01:47.ABCD}

Description: This task is used to initialize a frame counter
with a new value.

Arguments: Argument A specifies which counter and BCD
specifies the new counter value. No other
frame counters are affected unless there are no
arguments, then all frame counters are set to 00.
Upon reset all frame counters are set to 00.

Frame Counter Initialization Table

Data Field	New Counter Value			
	Proj#1	Proj#2	Proj#3	Proj#4
ABCD				
1078	78			
4140				140
3020			20	
2050		50		

Task Number:48

Function:Display frame counter values

Format:{01:48.}

Description: This task returns the frame counter packet to the host computer when it is executed.

Examples: Typical return status packet:

[045:101:004:086]

The projector numbers corresponding to the frame counter values are inferred by the positions in the packet.

[Projector 1:Projector 2:Projector3:Projector 4].

Task Number:49

Function:Delay loop

Format:{01:47.AB}

Description: This task specifies the number of second to pause before executing next task.

Arguments: The arguments are used to specify the number of seconds, up to 99 seconds, to pause before the next task is executed. Arguments are in decimal.

Delay Loop Table

Data Field (AB)	Time (sec)
00	No Operation
06	6
17	17

Task Number:4A

Function:Raise the projector screen

Format:{08:4A.AB}

Description: This task raises the projector screen for the number of seconds that are specified in arguments AB.

Arguments: The argument is in decimal. I.E. AB = 15 means 15 seconds. The arguments are used to specify the number of seconds, up to 99, to raise the screen. The format is identical to task 49. For the current project (7/91) 13 seconds works well.

NOTE the task node address in 08 not 01. This is a kluge based on the ease of adding the screen control hardware to the light controller box. Ideally all the projector tasks should be located at the same address.

Task Number:4B

Function:Lower the projector screen

Format:{08:4B.AB}

Description: This task lowers the projector screen for the number of seconds that are specified in arguments AB.

Arguments: The argument is decimal. I.E. AB = 15 means 15 seconds. The arguments are used to specify the number of seconds, up to 99 to lower the screen. The format is identical to task 49. For the current project (7/91) 13 seconds works

well.

NOTE the task node address in 08 not 01. This is a kluge based on the ease of adding the screen control hardware to the light controller box. Ideally all the projector tasks should be located at the same task node.

Light Controller Specific Tasks

The light control tasks are Task50 (Set delay) and Task51 (fade the lights). !Note! Task 51 must be a queued task, this is due to the lights having final fade values that are not limited to full on or full off.

Task Number:50

Function:Delay the start of next Task51

Format:{08:51.ABCDEFGH}

Description: This task is designed to specifically work as a front-end for Task51, the actual light fading task. This task loads a delay register, DLAYARY, with the four compacted arguments passed to it by the operating system. These arguments are used to set delays for the lights (1-4) that are connected to its node.
AB = lite1, CD = lite2, EF = lite3, GH = lite4.

The argument is the number of seconds to delay the start of the fade for each light when the next task51 is received. Note, as the code is now written, all four arguments must be used. If, say, light3 needs to be delayed 8 secs, but 1,2, or 3 need no delay. The packet would be {08:51.00000800}

Arguments: The arguments are all in hex. This is a little confusing and probably should be changed.

Data Field	Delay Times (sec)			
	LITE #1	LITE #2	LITE #3	LITE #4
03100700	3	16	7	0
0A0F080C	10	15	8	12
050A0F14	5	10	15	20

Examples:

{08:50.050A0f14} Load DLAYARY with the arguments for delays of 5sec, 10sec, 15sec, and 20secs for lights 1 to 4 respectively.

Task Number:51

Function:Control fade rates and intensities of the 4 lights

Format:{08:51.ABCDEFGHIJ} NOTE! Must be a queued task!

Description: This task is used to specify the fade rate of

the four lights and the final intensity levels.

The fifth argument is used as a dummy argument.

Its function is to flag task51 to use the values

stored in the array DLAYARY to set delays for

execution of the individual fade task for each

light. If five arguments are included, and a

task50 was not the previous task, you probably

will not like the results as Task51 will use

what ever it finds in DLAYARY and set delays

based on these values. Unlike the projector

task41, task51 allows final steady state light

intensities of variable brightness. The lights

can have final values from 00% to 100% intensity

that are in increments of 10%. The format for

the arguments are the same as for task41.

Correct assignment of task node and argument

number, to the associated light, is left up to

the user.

Arguments: The argument numbers correspond directly to the light numbers. Data fields A, C, E, and G specify the fade rates and data fields B, D, F, and H specify the final intensity levels for lights 1 to 4 respectively. The following lists the possible fade rates and intensity values:

Fade Rate Table (Arguments A,C,E,G)

Fade Value	Fade Rate (Secs)
0	Instantly
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	12
C	15
D	20
E	25
F	30

Final Intensity Level (Args B,D,F,H)

Intensity Value	Intensity Level
0	No change
1	100%
2	90%
3	80%
4	70%
5	60%
6	50%
7	40%
8	30%
9	20%
A	10%
B	00%
C	00%
D	00%
E	00%
F	00%

Examples: {08:51.314F} Queued task, fade lite1 to full on in 3 secs & fade lite2 to full off in 4 secs.

{08:51.00630741} Queued task, no change to lite1, lite2 to 80% in 6 secs, lite3 immediate 40%, lite4 to 100% in 4 secs.

{08:50.050A0f14} Load DLAYARY with the arguments for delays of 5sec, 10sec, 15sec, and 20secs for lights 1 to 4.

{08:51.5151515177} The fifth argument triggers the delay function with the values loaded in DLAYARY. The actions will, be upon entering the task: Pause 5 sec, Fade light1 to 100% in five secs, When it gets to full brightness, light2 will start to fade to 100% at the same rate.

Likewise for light3 and light4.